

# **Application Development in Web Mapping 3.**

## **WMS/WFS Server**

**László Kottván**

---

# Application Development in Web Mapping 3.: WMS/WFS Server

László Kottván

Lector: Antal Guszlev

This module was created within TÁMOP - 4.1.2-08/1/A-2009-0027 "Tananyagfejlesztéssel a GEO-ért" ("Educational material development for GEO") project. The project was funded by the European Union and the Hungarian Government to the amount of HUF 44,706,488.

v 1.0

Publication date 2010

Copyright © 2010 University of West Hungary Faculty of Geoinformatics

## Abstract

GeoServer allow users to share and edit geospatial data using several types of data sources and serving maps in a standard way. This Module gives an overview about GeoServer, OGC Web Map Service and Web Feature Service.

The right to this intellectual property is protected by the 1999/LXXVI copyright law. Any unauthorized use of this material is prohibited. No part of this product may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system without express written permission from the author/publisher.

---

# Table of Contents

3. WMS/WFS Server .....	1
1. 3.1 Introduction .....	1
2. 3.2 About GeoServer .....	1
2.1. 3.2.1 Installing GeoServer .....	1
2.1.1. 3.2.1.1 Installing Apache Tomcat .....	1
2.1.2. 3.2.1.2 Deploying GeoServer .....	4
2.2. 3.2.2 Data Directory .....	5
2.3. 3.2.3 Administration interface .....	6
2.4. 3.2.4 Data storing .....	7
3. 3.3 Working with layers .....	14
3.1. 3.3.1 Adding layers .....	14
3.2. 3.3.2 Viewing layers .....	17
3.3. 3.3.3 Grouping layers .....	18
4. 3.4 Web Map Service .....	19
5. 3.5 Web Feature Service .....	23



---

# Chapter 3. WMS/WFS Server

## 1. 3.1 Introduction

The Open Geospatial Consortium (OGC) in Web Map Service (WMS) and Web Feature Service (WFS) specifications defines the way how to serve geospatial raster and vector data to clients through networks, including the Internet.

There are several software providers which implement OGC Web Services (OWS) in their desktop or web products<sup>1</sup>.

One great open source server solution is the Java-based GeoServer.

In this module there will be:

- given instructions about GeoServer installation on Windows XP and Ubuntu 10.4 operating systems,
- reviewed the administration interface of GeoServer, including data management possibilities,
- added some layers to the server,
- demonstrated the geospatial data serving according to WMS and WFS.

## 2. 3.2 About GeoServer

GeoServer allow users to share and edit geospatial data using several types of data sources and serving maps in a standard way. Being a community-driven project, GeoServer is developed, tested, and supported by a diverse group of individuals and organizations from around the world.

GeoServer functions as the reference implementation of the Open Geospatial Consortium WFS, WMS standards and the WCS (Web Coverage Service) specification.

### 2.1. 3.2.1 Installing GeoServer

There are some ways to install GeoServer on an operating system. We will install it as a Web archive (WAR), a standalone servlet inside a servlet container. As a servlet container web server we will use Apache Tomcat which is an open source tool developed by the Apache Software Foundation. Tomcat requires Java Runtime Environment, you can download it from here: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

#### 2.1.1. 3.2.1.1 Installing Apache Tomcat

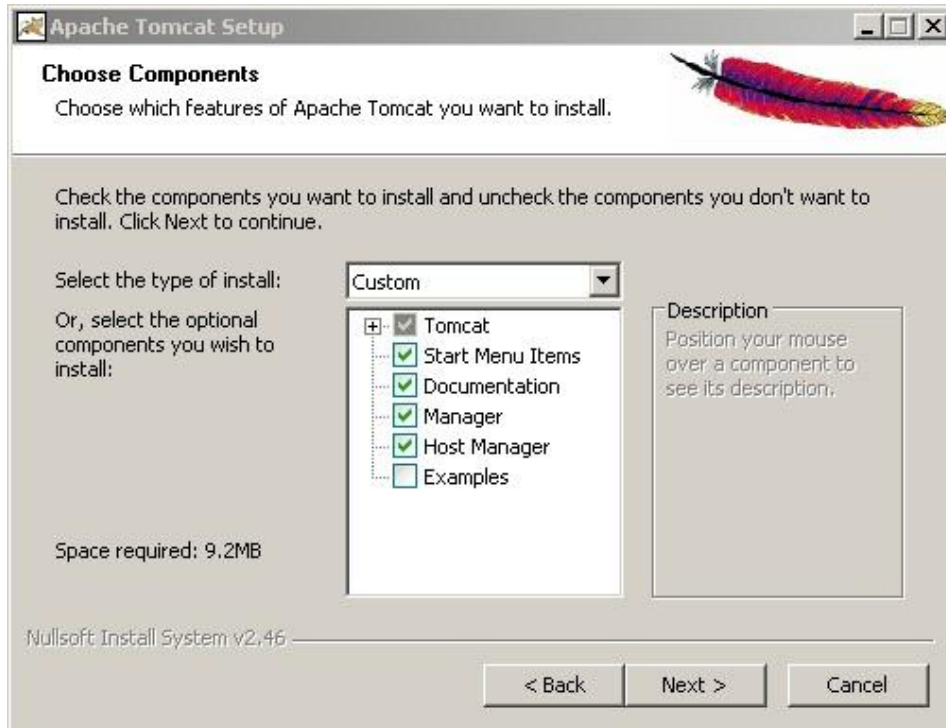
##### On Windows

Download the 32-bit/64-bit Windows Service Installer from <http://tomcat.apache.org/download-70.cgi>.

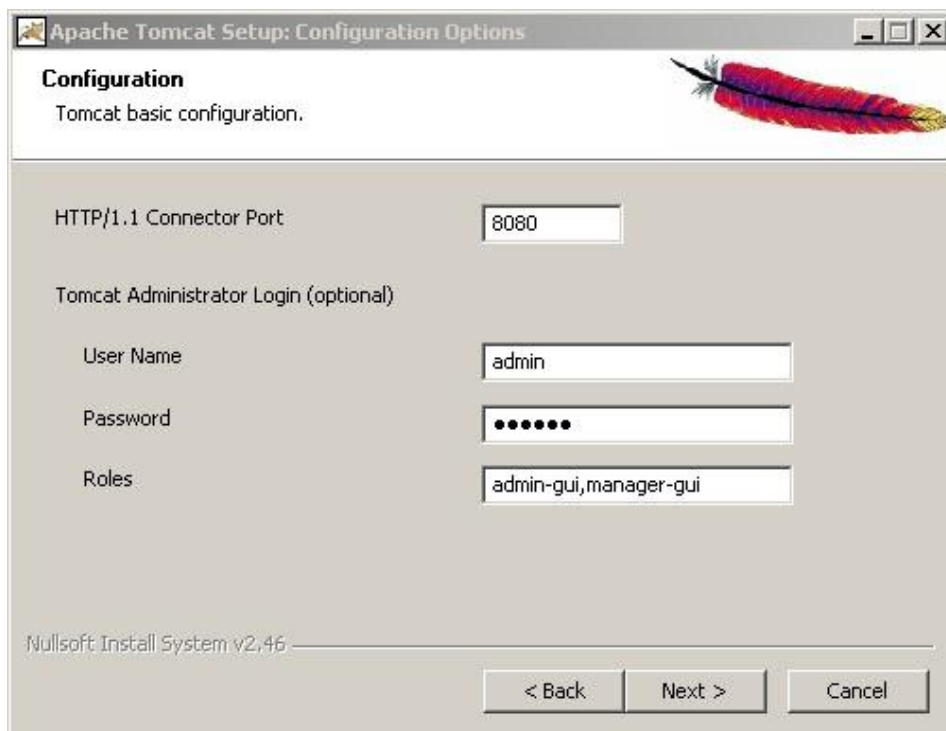
Run the installer and select the Normal installation or the Custom one. Manager is an important administrative tool to deploy applications. Host manager is for manage virtual hosts.

---

<sup>1</sup> OGC compliant softwares and services: <http://www.opengeospatial.org/resource/products>

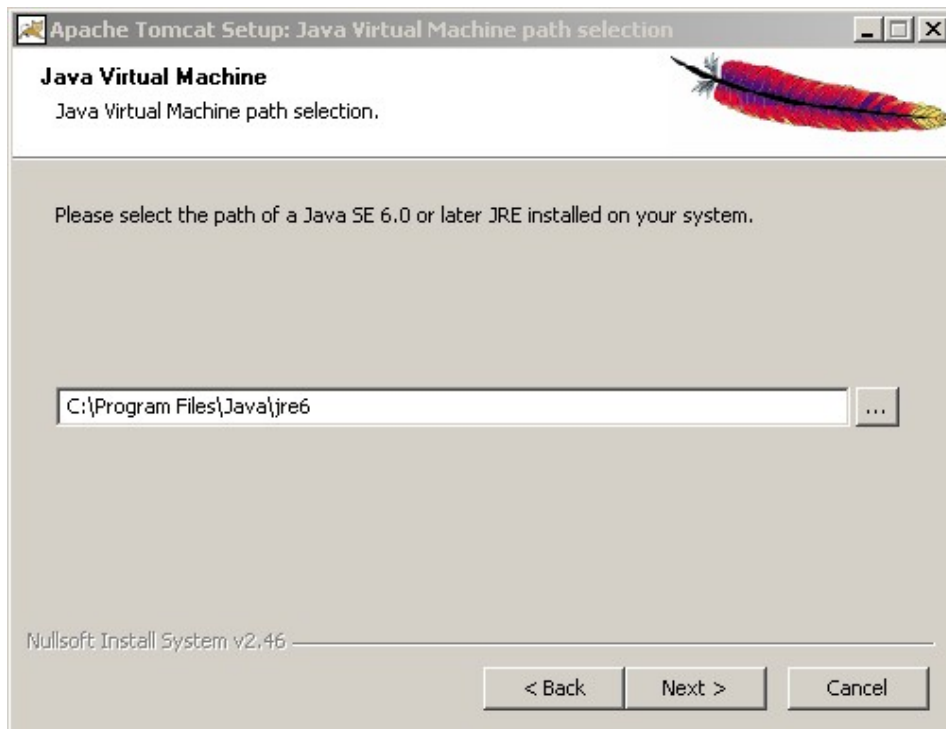


Type the administrator user and password.

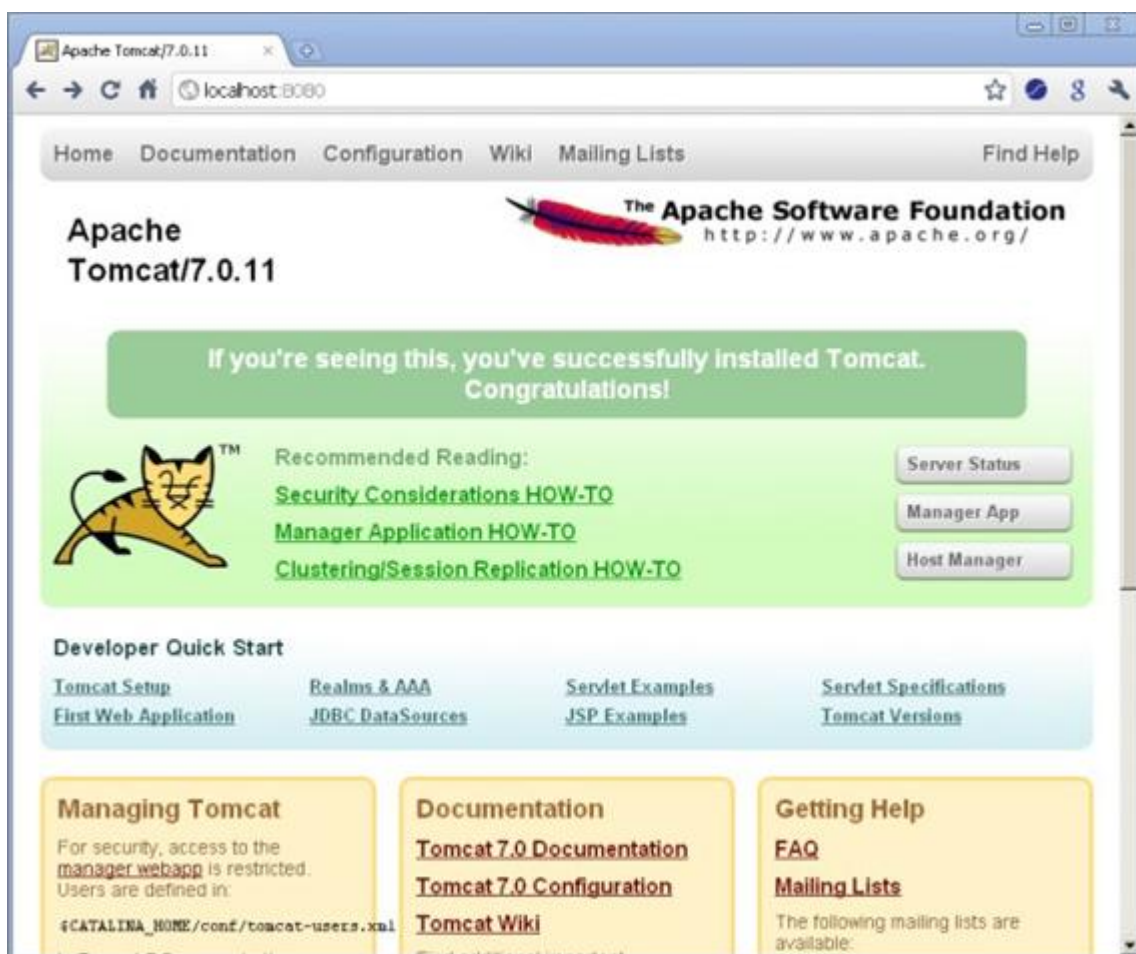


The default Tomcat port is 8080. The displayed roles are important to access for the Manager and Host manager applications.

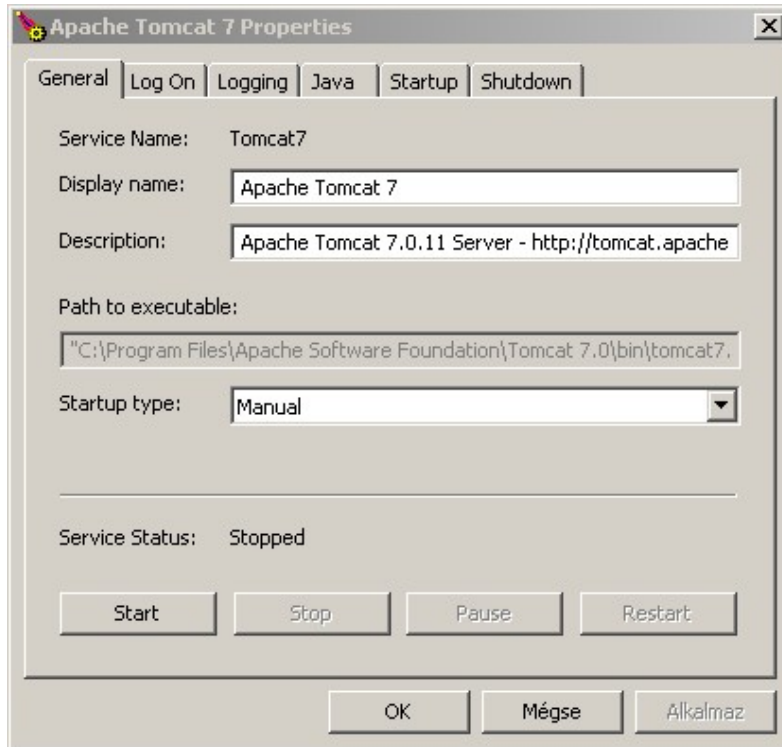
Select the path of the installed Java Virtual Machine.



If the installation is finished open `localhost:8080` in your browser. You should check your Manager and Host manager access.



From Start menu/Programs/Apache Tomcat 7.0 besides the documentations a configuration tool is available. With this tool you can set the properties of the server and start or stop it.



## On Ubuntu

Open a terminal and type:

```
sudo apt-get install tomcat6 tomcat6-admin tomcat6-common tomcat6-user tomcat6-docs tomcat6-examples
```

To enable admin web based features edit roles in `/etc/tomcat6/tomcat-user.xml`

Type:

```
gksudo gedit /etc/tomcat-user.xml
```

Add the following lines:

```
<role rolename="manager"/>
<role rolename="admin"/>
<user name="admin" password="secret_password" roles="manager,admin"/>
```

The "secret\_password" value should be changed to your secret password.

### 2.1.2. 3.2.1.2 Deploying GeoServer

Download the GeoServer in Web Archive format from: <http://geoserver.org/display/GEOS/Stable>.

Unzip it and you will have a geoserver.war file.

Log in to your Tomcat Manager application and roll down your browser to see WAR file to deploy section. Select the geoserver.war from your hard drive and deploy it.



Select WAR file to upload  geoserver.war

---

Open `http://localhost:8080/geoserver` and you should see the Welcome page of GeoServer.



GeoServer is installed in Tomcat/webapps directory. For example on Windows the path looks like this:

`C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\geoserver`

To log in the administrative area the default settings are:

- username: admin
- password: geoserver

## 2.2. 3.2.2 Data Directory

The data directory stores all configurations including data sources, services, metadata and more. The data directory path is `geoserver/data`. On Windows the full path looks like this:

`C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps\geoserver\data`

The GeoServer data directory has the following structure:

- `global.xml` - contains general service settings and contact information
- `logging.xml` - sets the logging parameters
- `wms.xml` - contains WMS service metadata and settings
- `wfs.xml` - contains WFS service metadata and settings
- `wcs.xml` - contains WCS service metadata and settings
- `data/` - used to store shp files and raster files
- `demo/` - contains files for demonstrating sample requests
- `gwc/` - store the cached data

- `layergroups/` - contains information about layergroups configurations
- `palettes/` - store image palettes for WMS in order to better serving
- `plugIns/` - contains plugin definitions
- `security/` - contains files to configure the GeoServer security subsystem
- `styles/` - contains Styled Layer Descriptor files
- `templates/` - contains files used by the GeoServer templating subsystem
- `user-projections/` - contains `epsg.properties` file which is used to define custom spatial reference systems
- `workspaces/` - contains metadata about layers
- `www/` - is used to allow GeoServer to act like a regular web server and serve regular files

### **2.3. 3.2.3 Administration interface**

The GeoServer administration interface is available after logging in. The left side of the interface contains the following main elements:

- **Server** - provides access to GeoServer environment information and enable some configurations
- **Services** - configuration of WCS, WFS, WMS services
- **Data** - adding, editing, deleting different data types and organize them into GeoServer data structures
- **Security** - for setting user, data and service security parameters
- **Demos** - contains links to example requests and a list of known SRS (Spatial Reference System), it is available without logging in.
- **Layer Preview** - provides layer views in various output formats



For more information about administration read the related pages of GeoServer User Manual<sup>2</sup>.

## 2.4. 3.2.4 Data storing

GeoServer manages the following data types:

- Vector data formats
  - Shapefiles
  - PostGIS databases
  - Java Properties files
- Raster data formats
  - ArcGrid
  - GeoTIFF
  - Gtopo30

---

<sup>2</sup> GeoServer User Manual, Web Administration Interface: <http://docs.geoserver.org/stable/en/user/webadmin/index.html>

- ImageMosaic
- WorldImage
- Other data formats
  - External WMS layers
  - External WFS layers

Some data types are available through extensions and require additional installation steps:

- ArcSDE
- GML
- DB2
- H2
- MySQL
- Oracle
- Microsoft SQL Server
- Pregeneralized Features
- Vector Product Format
- GDAL program for additional image formats
- Image Pyramid extension for pyramid representation<sup>3</sup>
- Image Mosaic JDBC
- Oracle Georaster
- Custom JDBC Access for image data

Data configuration requires to set:

- the properties of data (in `Layers`),
- the location of the data source (in `Stores`),
- and a namespace<sup>4</sup>, a logical group of data sets (in `Workspaces`).

Let's see these configuration options through an example data.

### Workspace

Selecting the `Workspaces` from the left side of administration interface you will get a list of defined workspaces and there is an add and a remove option on the top.

Clicking on `topp` workspace the `Edit Workspace` form is displayed with Name, Namespace URI and Default Workspace properties.

---

<sup>3</sup>[http://en.wikipedia.org/wiki/Pyramid\\_\(image\\_processing\)](http://en.wikipedia.org/wiki/Pyramid_(image_processing))

<sup>4</sup><http://en.wikipedia.org/wiki/Namespace>

## Edit Workspace

Edit existing workspace

Name

Namespace URI

The namespace uri associated with this workspace

Default Workspace

☐

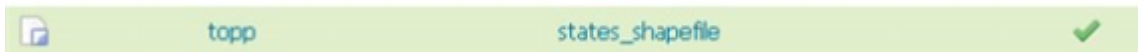
The namespace can be set related to the data provider, data owner, a project or any logical category which is somehow associated to the datasets in the workspace.

### Stores

A store is the name for a container of geographic data. A store refers to a specific data source.

Selecting the `Stores` the given list tells the available stores in the defined workspaces. Each store must be associated with one workspace.

For example `topp` workspace contains `states_shapefile` store.



Clicking on `states_shapefile` an edit form is displayed with Basic Store Info and Connection Parameters. Here the namespace can be modified.

## Edit Vector Data Source

Shapefile

ESRI(tm) Shapefiles (\*.shp)

### Basic Store Info

Workspace \*

Data Source Name \*

Description

☒ Enabled

### Connection Parameters

URL \*

Namespace \*

☐ create spatial index

charset

☐ memory mapped buffer



The URL refers to the location of data source in data dictionary.

file:data refers to a data folder in data directory (geoserver/data/data).

In this example, according to the URL, the states shapefile path is:

```
geoserver/data/data/shapefiles/states.shp
```

### Layers

A layer is a collection of geospatial features or a coverage. Vector layer is analogous to feature type and raster layer is analogous to coverage. Usually a layer contains one type of data (points, lines, polygons, raster). A store can contain many layers but it must contain at least one layer. (It has a single layer in the case of a shapefile.)

Selecting Layers a list displays the layers with layer type, layer name, store name, workspace name, status and SRS.

	topp	states_shapefile	states		EPSG:4326
---	------	------------------	--------	---	-----------

Clicking on states the layer can be edited in Data and Publishing tabs.

## topp:states

Configure the resource and publishing information for the current layer

**Data****Publishing**

---

### Basic Resource Info

**Name****Title****Abstract**

This is some census data on the states.

---

### Keywords

**Current Keywords**

census

united

boundaries

state

states

**New Keyword**

---

### Metadata links

No metadata links so far

- Name - used to reference the layer in WMS requests.
- Title - a human-readable name of the layer for clients.
- Abstract - a description about the layer.
- Keywords - words associated with the layer to aid in catalog searching.
- Metadata links - linked external documents that describe the layer.

### Coordinate Reference Systems

#### Native SRS

[GCS\\_WGS\\_1984...](#)

#### Declared SRS

[EPSG:WGS 84...](#)

#### SRS handling

▼

### Bounding Boxes

#### Native Bounding Box

Min X	Min Y	Max X	Max Y
<input type="text" value="-124,731"/>	<input type="text" value="24,956"/>	<input type="text" value="-66,97"/>	<input type="text" value="49,372"/>

[Compute from data](#)

#### Lat/Lon Bounding Box

Min X	Min Y	Max X	Max Y
<input type="text" value="-124,731"/>	<input type="text" value="24,956"/>	<input type="text" value="-66,97"/>	<input type="text" value="49,372"/>

[Compute from native bounds](#)

### Feature Type Details

Property	Type	Nullable	Min/Max Occurences
the_geom	MultiPolygon	true	0/1
STATE_NAME	String	true	0/1
STATE_FIPS	String	true	0/1
SUB_REGION	String	true	0/1
STATE_ABBR	String	true	0/1

- Native SRS - the projection of the stored layer
- Declared SRS - the projection what GeoServer gives to clients
- SRS Handling - determines how GeoServer should handle the different projections
- Native Bounding Box - determines the extent of the layer, can be generated
- Lat/Lon Bounding Box - computed bounds based on the standard lat/long, can be generated
- Feature Type Details:
  - property and type of data,
  - Nullable is true if the value can be empty, null
  - how many values allowed to have



## topp:states

Configure the resource and publishing information for the current layer

Data	Publishing
------	------------

---

### Basic Settings

**Name**

☒ Enabled

### HTTP Settings

☒ Response Cache Headers

**Cache Time (seconds)**

### WFS Settings

**Per-Request Feature Limit**



**Maximum number of decimals**

### WMS Settings

**Default Style**

☒ < 2M  
☐ 2M - 4M  
☐ > 4M  
☐ Boundary

**Additional Styles**

Available Styles		Selected Styles
burg capitals cite_lakes dem giant_polygon grass green line poi point	 	polygon pophatch

**Default Rendering Buffer**

**Default WMS Path**

- HTTP Settings - if Response Cache Headers is checked, Geoserver will not request the same tile twice within the time specified in Cache Time
- WFS Settings - the maximum number of features a WFS GetFeature operation should generate

- Default style - the style that will be used when the client does not specify it
- Additional styles - other styles that can be associated to this layers for alternative styling
- Default rendering buffer - the default value of the buffer GetMap/GetFeatureInfo vendor parameter

- Default WMS path - the location of the layer in the WMS capabilities layer tree

#### WMS Attribution

Attribution Text

Attribution Link

Logo URL

Logo Content Type

Logo Image Width

Logo Image Height

[Auto-detect image size and type](#)

- WMS Attribution - information about data providers

#### KML Format Settings

Default Regionating Attribute

Default Regionating Method

Features Per Regionated Tile

Save

Cancel

- KML Format Settings - Regionation<sup>5</sup> settings to decide how to determine which features show up more prominently than others, it can be useful for visualizing large datasets. GeoServer can serve KML data applying a regionating method on an attribute or geometry.

## 3. 3.3 Working with layers

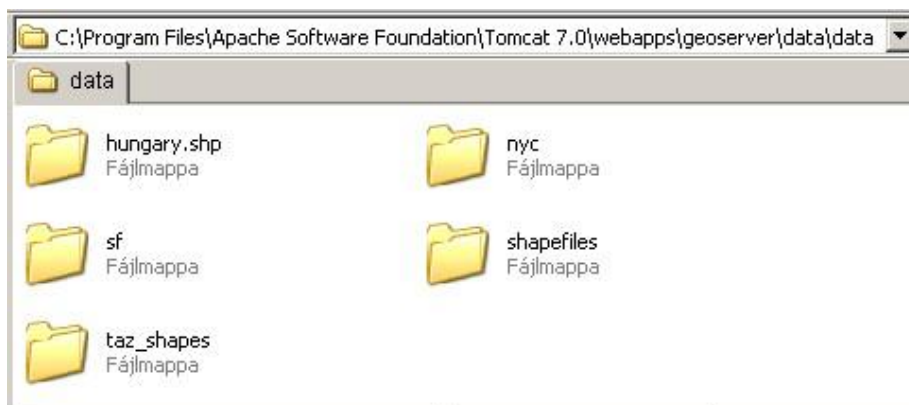
### 3.1. 3.3.1 Adding layers

From <http://download.geofabrik.de/osm/europe/> download the `hungary.shp.zip` file. It contains seven shapefiles which will be used for adding layers to GeoServer.

Unzip it and move it to `geoserver/data/data` folder.

---

<sup>5</sup>Regionating Vector Data:<http://earth.google.com/support/bin/answer.py?answer=175007>



Create a new workspace, for example:

- Name: AWD
- Namespace URI: <http://geo.info.hu/awd>

## New Workspace

Configure a new workspace

**Name**

**Namespace URI**

The namespace uri associated with this workspace

**Default workspace**

☐

**Submit**







**Cancel**

Add a new store, using the [Directory of spatial files](#) link. We need this because the hungary.shp contains several shapefiles.






## New data source

Choose the type of data source you wish to configure

### Vector Data Sources

-  **Directory of spatial files** - Takes a directory of spatial data files and exposes it as a data store
-  **PostGIS** - PostGIS Database
-  **PostGIS (JNDI)** - PostGIS Database (JNDI)
-  **Properties** - Allows access to Java Property files containing Feature information
-  **Shapefile** - ESRI(tm) Shapefiles (\*.shp)
-  **Web Feature Server** - The WFSDataStore represents a connection to a Web Feature Server, ability to perform transactions on the server (when supported / allowed).

### Raster Data Sources

-  **ArcGrid** - Arc Grid Coverage Format
-  **GeoTIFF** - Tagged Image File Format with Geographic information
-  **Gtopo30** - Gtopo30 Coverage Format
-  **ImageMosaic** - Image mosaicking plugin
-  **WorldImage** - A raster file accompanied by a spatial data file

Select AWD workspace from the workspace list, check **Enabled** and enter the followings:

- **Data Source Name:** hungary\_shapes
- **Description:** Hungary from OSM
- **URL:** file://data/hungary.shp

## Edit Vector Data Source

Directory of spatial files

Takes a directory of spatial data files and exposes it as a data store

### Basic Store Info

**Workspace \***

AWD

**Data Source Name \***

hungary\_shapes

**Description**

Hungary from OSM

☒ **Enabled**

### Connection Parameters

**URL \***

file:data/hungary.shp

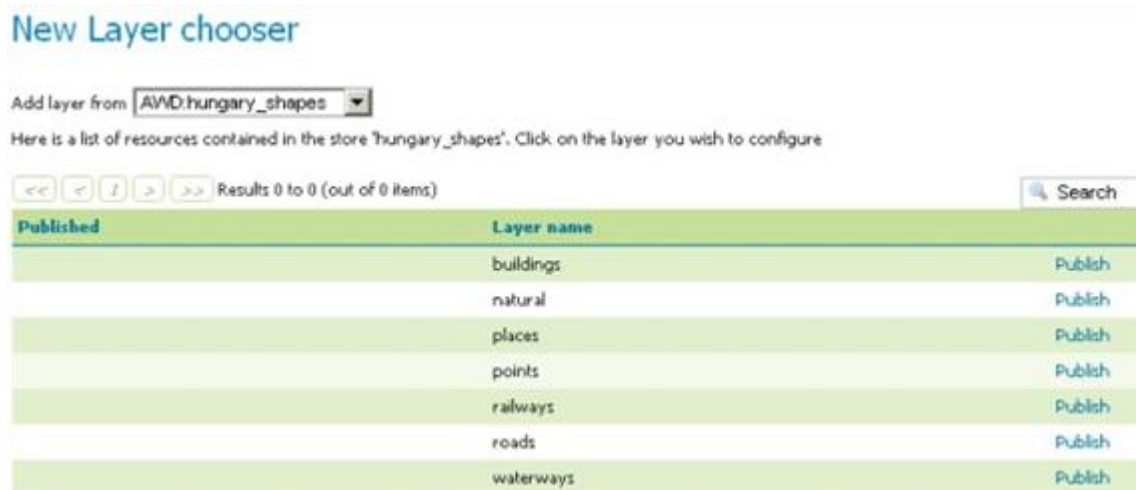
**Namespace \***

http://geo.info.hu/awd

**Save**

**Cancel**

Add layers choosing first the AWD:hungary\_shapes and the Publish link of buildings on New Layer chooser page.



On data tab of the displaying form:

- enter a description
- enter keywords to the selected layer
- find the EPSG:4326 SRS value from SRS list or enter manually to Declared SRS field (Native SRS is probably unknown and you cannot change it)
- compute bounding box values and save the form data
- repeat these steps with the other layers in hungary\_shapes store.



Finally you should have seven hungary\_shapes layers in layers list.

### 3.2. 3.3.2 Viewing layers

GeoServer provides a great built in solution to checking served layers in different output format.

Select Layer Preview and look at the created layers in different formats. You can choose the Common Formats (OpenLayers, KML, GML) or Select one from the list.



### 3.3.3.3 Grouping layers

Creating a group from layers is easy. GeoServer calculate the bounds and provides options to add layers from any stores and declare the order of them.

**Layer group**

Edit the contents of a layer groups

**Name**

hungary\_layer\_group

**Bounds**

Min X	Min Y	Max X	Max Y
16,114	45,743	22,901	48,586

EPSG:4326 [Find...](#) EPSG:WGS 84...

[Generate Bounds](#)

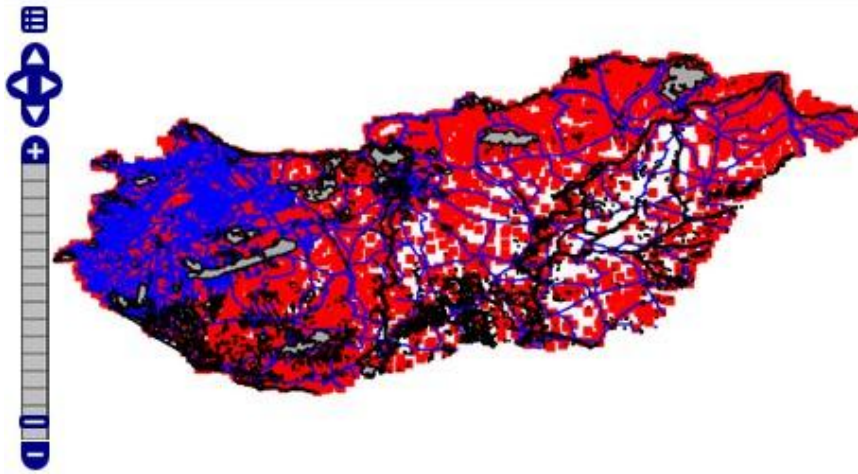
[Add Layer...](#)

**Layers**

Position	Layer	Default Style
↓	AWD:places	<input checked="" type="checkbox"/>
↑ ↓	AWD:railways	<input checked="" type="checkbox"/>
↑ ↓	AWD:waterways	<input checked="" type="checkbox"/>
↑	AWD:natural	<input checked="" type="checkbox"/>

Styling can be done with the default style or by selecting an alternate style.

The output with default styles:



Styling in GeoServer is based on Styled Layer Descriptor (SLD) OGC specification. An SLD file can be created using the GeoServer built-in editor which provides a validation also. To get more information about styling see AWD Module 5 or read the User Manual<sup>6</sup>.

## 4. 3.4 Web Map Service

A Web Map Service (WMS) produces maps of spatially referenced data dynamically from geographic information. WMS-produced maps are generally rendered in PNG, GIF, JPEG, SVG and WebCGM formats.

To work with maps, the standard defines three operations:

- GetCapabilities
- GetMap
- GetFeatureInfo

This operations used in HTTP is requested by clients. Each operation produces a response in a particular format. WMS server which supports SLD (Styled Layer Descriptor) generally has DescribeLayer and GetLegendGraphic operations.

A HTTP GET request is an URL which has a query part after the ? character. The query has parameters separated with & sign. The parameters are given with name and value separated with = sign. Here is an example:

```
http://www.example.com/myserver?param1=value1&param2=value2
```

GeoServer supports both WMS 1.1.1 and 1.0.3 standards. In GetMap request the srs parameter from 1.1.1 in 1.0.3 is changed to crs. The axis ordering from longitude/latitude or x/y in 1.1.1 in 1.0.3 is changed to latitude/longitude or y/x.

In GetFeatureInfo request the x, y parameters from 1.1.1 in 1.0.3 is changed to i, j.

### GetCapabilities

This operation is used to get information about what operations and services are offered by a server. The GetCapabilities parameters are:

---

<sup>6</sup> GeoServer User Manual, Styling section: <http://docs.geoserver.org/stable/en/user/styling/index.html>

Request parameter	Mandatory/optional	Description
VERSION=version	O	Request version
SERVICE=WMS	M	Service type
REQUEST=GetCapabilities	M	Request name
FORMAT=MIME_type	O	Output format of service metadata
UPDATESEQUENCE=string	O	Sequence number or string for cache control

Let's get the capabilities of the local GeoServer. Make sure that your server is running, open a browser window and paste:

```
http://localhost:8080/geoserver/wms?service=wms&request=GetCapabilities
```

Notes:

- The request in GeoServer works without the service parameters too.
- In GeoServer the default route of WMS is /geoserver/wms and the route of WFS is /geoserver/wfs.

The response is an XML document. It can be opened in an editor or Web browser. The first part of it is in <Service> tag the second part is in <Capability> tag. The Service contains information about the WMS service and the service contact who is responsible for the service. These information can be configured on Administration interface Server/Contact Information and Services/WMS pages.

The Capability contains metadata about map formats, layers, legends, SRSs.

Here is a part of the XML document that describes the natural layer from hungary\_shapes:

```
<Layer queryable="1">
  <Name>AWD:natural</Name>
  <Title>natural</Title>
  <Abstract/>
  <KeywordList/>
  <SRS>EPSG:4326</SRS>
  <LatLonBoundingBox minx="16.12" miny="45.744" maxx="22.901" maxy="48.585"/>
  <BoundingBox SRS="EPSG:4326" minx="16.12" miny="45.744" maxx="22.901"
maxy="48.585"/>
  <Style>
    <Name>polygon</Name>
    <Title>Default Polygon</Title>
    <Abstract>A sample style that draws a polygon</Abstract>
    <LegendURL width="20" height="20">
      <Format>image/png</Format>
      <OnlineResource
xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="http://localhost:8080/geoserver/wms?request=GetLegendGraphic&
format=image%2Fpng&width=20&height=20&layer=natural"/>
      </LegendURL>
    </Style>
</Layer>
```

## GetMap

The GetMap request parameters are:



Request parameter	Mandatory/optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetMap	M	Request name.
LAYERS=layer_list	M	Comma-separated list of one or more map layers.
STYLES=style_list	M	Comma-separated list of one rendering style per requested layer.
CRS=namespace:identifier	M	Coordinate reference system.
BBOX=minx,miny,maxx,maxy	M	Bounding box corners (lower left, upper right) in CRS units.
WIDTH=output_width	M	Width in pixels of map picture.
HEIGHT=output_height	M	Height in pixels of map picture.
FORMAT=output_format	M	Output format of map.
TRANSPARENT=TRUE FALSE	O	Background transparency of map (default=FALSE).
BGCOLOR=color_value	O	Hexadecimal red-green-blue colour value for the background color (default=0xFFFFFF).
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default=XML).
TIME=time	O	Time value of layer desired.
ELEVATION=elevation	O	Elevation of layer desired.
Other sample dimension(s)	O	Value of other dimensions as appropriate.

Remember in WMS 1.1.1 there is SRS parameter not CRS.

GeoServer output formats:

- PNG - parameter syntax: `format=image/png`
- PNG8 - 8 bit (256 colors) image, parameter syntax: `format=image/png8`
- JPEG - parameter syntax: `format=image/jpeg`
- GIF - parameter syntax: `format=image/gif`
- TIFF - parameter syntax: `format=image/tiff`
- TIFF8 - 8 bit image, parameter syntax: `format=image/tiff8`
- GeoTIFF - includes GeoTIFF metadata, parameter syntax: `format=image/geotiff`
- GeoTIFF - 8 bit image, parameter syntax: `format=image/geotiff8`
- SVG - parameter syntax: `format=image/svg`
- PDF - parameter syntax: `format=application/pdf`
- GeoRSS - parameter syntax: `format=rss`
- KML - parameter syntax: `format=kml`
- KMZ - parameter syntax: `format=kmz`
- OpenLayers - generates an OpenLayers client application, parameter syntax: `format=application/openlayers`

Using the data from GetCapabilities XML we can compose a request like this:

```
http://localhost:8080/geoserver/wms?
version=1.1.1&
request=GetMap&
layers=AWD:natural&
```

```
styles=polygon&
srs=EPSG:4326&
bbox=16.12,45.744,22.901,48.585&
width=600&
height=400&
format=image/png
```

GeoServer offers an easier solution too. With WMS Reflector<sup>7</sup> the request can be shortened. WMS reflector adds some default values to the request. These are the following:

- request=getmaps
- version=1.1.1
- format=image/png
- width=512
- height=512, if width is not specified
- EPSG:4326
- bbox - the bounds of layers

An example reflector request:

```
http://localhost:8080/geoserver/wms/reflect?
layers=AWD:natural&
styles=polygon&
width=600&
height=400
```

### GetFeatureInfo

The GetFeatureInfo operation is more information about features on the map images that were returned by previous Map requests. The general use case for GetFeatureInfo is that a user sees the response of a Map request and chooses a point on that map for which to obtain more information.

Request parameter	Mandatory/optional	Description
VERSION=1.3.0	M	Request version.
REQUEST=GetFeatureInfo	M	Request name.
map request part	M	Partial copy of the Map request parameters that generated the map for which information is desired.
QUERY_LAYERS=layer_list	M	Comma-separated list of one or more layers to be queried.
INFO_FORMAT=output_format	M	Return format of feature information (MIME type).
FEATURE_COUNT=number	O	Number of features about which to return information (default=1).
I=pixel_column	M	<i>i</i> coordinate in pixels of feature in Map CS.
J=pixel_row	M	<i>j</i> coordinate in pixels of feature in Map CS.
EXCEPTIONS=exception_format	O	The format in which exceptions are to be reported by the WMS (default= XML).

Remember in WMS 1.1.1 there are x, y instead of i and j parameters.

An example request:

```
http://localhost:8080/geoserver/wms?
bbox=16.12,45.744,22.901,48.585&
styles=polygon&
format=png&
```

<sup>7</sup> WMS Reflector: <http://docs.geoserver.org/stable/en/user/tutorials/wmsreflector.html#tutorials-wmsreflector>

```
info_format=text/plain&
request=GetFeatureInfo&
layers=AWD:natural&
query_layers=AWD:natural&
width=600&
height=400&
x=170&
y=160
```

The result is a text document with feature information.

Read more about WMS in the specification document<sup>8</sup> or on the related pages of GeoServer User Manual<sup>9</sup>.

## 5. 3.5 Web Feature Service

The Web Feature Service (WFS) is a standard to send and receive geospatial data through HTTP. WFS transfers information in Geographic Markup Language (GML) format. GML is an OGC standard<sup>10</sup> to express geographic data in XML.

WFS operations are:

- GetCapabilities
- DescribeFeatureType
- GetFeature
- LockFeature
- Transaction
- GetGMLObject (in WFS 1.1.0 only)

Transactions can contain INSERT, UPDATE or DELETE elements to add, modify or remove features.

GeoServer supports WFS 1.0.0 and 1.1.0 standards.

### GetCapabilities

The GetCapabilities request gives an XML document with metadata of service, operations, feature types, functions (spatial, logical, arithmetic) which are provided by a server.

Insert the next request into your browser and study the result xml:

```
http://localhost:8080/geoserver/wfs?
service=wfs&
version=1.1.0&
request=GetCapabilities
```

### DescribeFeatureType

DescribeFeatureType returns an XML schema document describing the structure of the data set. The schema document lists each field and its data type.

```
http://localhost:8080/geoserver/wfs?service=WFS&
version=1.1.0&
request=DescribeFeatureType&
typeName=AWD:railways
```

---

<sup>8</sup> OGC WMS standard: <http://www.opengeospatial.org/standards/wms>

<sup>9</sup> GeoServer User Manual, Web Map Service: <http://docs.geoserver.org/stable/en/user/services/wms/index.html>

<sup>10</sup> OGC GML standard: <http://www.opengeospatial.org/standards/gml>

The result is an XML document.

### GetFeature

To get the geospatial data we can use the GetFeature request.

```
http://localhost:8080/geoserver/wfs?service=WFS&
version=1.1.0&
request=GetFeature&
typeName=AWD:railways
```

The result is a GML document.

Read more about WFS in the specification document<sup>11</sup> or on the related pages of GeoServer User Manual<sup>12</sup>.

To study some example requests, open your server Demos page and select Demo requests link.

## Bibliography

GeoServer: *GeoServer User Manual*, 2011, <http://docs.geoserver.org/stable/en/user/>

Scott, D.: *GIS for Web developers, Adding Where to Your Web Applications*, The Pragmatic Programmers LLC, 2007

---

<sup>11</sup> OGC WFS standard: <http://www.opengeospatial.org/standards/wfs>

<sup>12</sup> GeoServer User Manual, Web Feature Service: <http://docs.geoserver.org/stable/en/user/services/wfs/index.html>